AMENDMENTS TO SPECIFICATION

Page 1, lines 6-19:

The present invention relates to a high-speed search method for ~~a~~ an LSP (Local Spectrum Pair) using SVQ (Split Vector Quantization) and a fixed codebook of the G.729 speech encoder, and more particularly to a high-speed search method which may decrease overall computational complexity without sacrificing spectral distortion performance by reducing a size of the codebook using an order character of LSP ~~counts~~ parameters in searching a codebook having high computational complexity during quantizing a split vector of LSP ~~counts~~ parameters of a speech encoder, used to compress voice signals in a low speed, and a high-speed search method which may dramatically reduce computational complexity without loss of tone quality by detecting and searching tracks on the basis of a magnitude order of a correlation signal (d'(n)), obtained by ~~a~~ an impulse response and a target signal in the process of searching the fixed codebook of the G.729 speech encoder.

Page 2, lines 3-6:

For high-quality encoding, the low transmission speech encoder ~~quanitizes~~ quantizes LPC ~~counts~~ coefficients, in which an optimal LPC ~~count~~ coefficient is obtained by dividing the input speech signal in a frame unit to minimize predictive error energy in each frame.

Page 2, lines 8-12:

In the above conventional method, more bits should be assigned to quantize the 10 LPC ~~counts~~ coefficients. However, when directly quantizing the LPC ~~counts~~ coefficients, there are problems that characters of the filters are very sensitive to the quantization error and that stability of the LPC filter is not assured after quantizing the ~~counts~~ coefficients.

Page 2, lines 15-17:

Therefore, the present invention is designed to overcome the problems of the prior art. An object of the present invention is to provide a high speed search method for a speech encoder

having decreased overall computational complexity, and in which spectral distortion performance is not sacrificed.

Page 3, line 20 to Page 4, line 10:

Quantizing overall vectors at one time is substantially impossible because a size of the vector table becomes too big and too much time is taken for search. To solve this problem, the present invention employs SVQ (Split Vector Quantization) to divide overall vectors into several sub-vectors and then quantize the sub-vectors independently. A predictive SVQ, which is a method adding a prediction unit to the SVQ, uses correlation between frames of the LSP (Linear Spectrum Pair) ~~counts~~ parameters for more efficient quantization. That is, the predictive SVQ does not quantize the LSP of a current frame directly, but ~~predict~~ predicts the LSP of the current frame on the basis of ~~a~~ an LSP of the previous frame and then ~~quantize~~ quantizes a prediction error. The LSP has a close relation with a frequency character of the speech signal, ~~so~~ making time prediction possible with great gains.

Page 4, line 11 to Page 5, line 3:

When quantizing the LSP ~~counts~~ parameters with such VQ, most of quantizers have a ~~great amount of~~ large LSP codebook. And, in order to reduce computational complexity in searching an optimal code vector in the codebook, the quantizer decreases a range of codes to be searched by using an order of the LSP counts. That is, the quantizer arranges the code vectors in the codebook for a target vector in a descending order according to element values in a specific ~~row~~ position in a sub-vector. Then, the optimal code vector, which minimizes distortion in the arranged codebook, has nearly identical value with that of the target vector, which implies that such value has an order character. Under such presumption, the present invention compares an element value of a specific ~~row~~ position arranged in a descending order with element values of other adjacent ~~rows~~ positions, and then calculates distortion with high computational complexity for the code vectors, which satisfies the order character, and cancels the calculation process for other code vectors.

Page 5, lines 6-14:

FIG. 1 shows a structure of a general SVQ. As shown in the figure, the target vector, or LSP vector (p) satisfies the below order character.

[Equation 1]

$$\mathbf{0 < p_1 < p_2 < .... < p_p < \pi}$$

[Equation 2]

$$\mathbf{E_{l,m} = (p_m - p\}_{l,m}\}^T W_m(p_m - p\}_{l,m})}$$

$0 \le m \le M - 1$

$1 \le l \le L_m$

where l,m in the subscript of $E_{l,m}$ are indices that represent the *l*th index of the *m*th codebook, *i.e.*, the letters "*l*" and "m," and

where superscript T designates the transpose of $(\mathbf{p}_m - \mathbf{p}\}_{l,m}\}$ for purposes of determining the dot product of $(\mathbf{p}_m - \mathbf{p}\}_{l,m}\}$ and $\mathbf{W}_m (\mathbf{p}_m - \mathbf{p}\}_{l,m}\}$ in order to calculate the least-mean-square error $E_{l,m}$.

Page 5, line 22 to Page 6, line 6:

In the Equation 2, the LSP code vector ($\mathbf{p}\}$) is divided into M number of sub-vectors, each of which consists of L_m number of code vectors. Codebook magnitudes (L_o, L_1, ..., L_{M-1}) of M number may be assigned to a specific sub-vector to improve tone quality. $\mathbf{W}_m$ is a weighting matrix for the m[th] sub-vector and obtained by a non-quantized ~~LSF~~ LSP vector (**p**).

Page 6, lines 7-20:

In order to employ a high-speed search method in the present invention, conversion of the conventional codebook is needed. This is a process of replacing the conventional codebook with a new codebook having L reference rows, as illustrated in Fig. 3, which is arranged in a descending order on the basis of a specific row (or, reference row), experimentally determined. The reference row is selected for each codebook and should be a row in which an average search range is minimized experimentally. The average search range is an average number with which an element value of the n[th] row in the arranged codebook based on each n[th] row and an element

value of n + 1th and n - 1th positions in the target vector satisfy the order character with use of the target vector for the arranged codebook.

[Equation 3]

$\mathbf{p}\}_{l,n} > \mathbf{p}\}_{n-1},\quad 1 \le l \le L,\quad 0 \le n \le 8$

$\mathbf{p}\}_{l,n} > \mathbf{p}\}_{n+1},\quad 1 \le l \le L,\quad 1 \le n \le 9$

where l,n in the subscript of $\mathbf{p}_{l,n}$ are indices that represent the *l*th index of the *n*th reference row, *i.e.*, the letters "*l*" and "n."

Page 11, line 12 to Page 12, line 1:

An efficient search method of the fixed codebook is very important for high quality speech encoding in a low-transmission speech encoder. In the G.729 speech encoder, the fixed codebook is searched for each sub-frame, and 17-bit logarithmic codebook is used for the fixed codebook and an index of the searched codebook is transmitted. A Vector in each fixed codebook has 4 pulses. As shown in Table 1, each pulse has ~~with~~ a size of +1 or -1 in a designated position ~~as shown in Table 1~~ and is represented ~~like~~ by the Formula 6.

[Equation 6]

$$c(n) = \sum_{i=0}^{3} s_i \delta(n - m_i) \qquad n = 0, 1, \ldots\ldots, 39$$

in which **c(n)** is a fixed codebook vector, ~~and~~ **δ(n)** is a unit pulse, and $\mathbf{m}_i$ is a position of the ith pulse.

Page 12, lines 11-13:

Assuming that a codebook vector of an index (k) is c_k, an ~~optical~~ optimal code vector is selected as a codebook vector, which maximizes the following Formula 8.

Page 13, lines 5-8:

The codebook search is comprised of 4 loops, each of which determines a new pulse. The matrix C_k that is squared in the numerator ~~in the~~ of Formula 8 is given ~~as~~ by C in the following

Formula 11, and the denominator in the Formula 8 is given as the following Formula 12 (in which $\phi(m_i, m_j)$ corresponds to $\Phi(i,j)$ of equation 10).

Page 13, lines 16 to Page 3:

In order to reduce the computational complexity in the codebook search, the following process is employed. At first, d(n) is decomposed into an absolute value $d'(n) = | d(n) |$ and its sign. At this time, the sign value is previously determined for the available 40 ~~pulses~~ pulse positions in Table 1. And, the matrix Φ is modified into $\phi'(i,j) = \text{sign}[s(i)]\ \text{sign}[s(j)]\ \phi(i,j)$, $\phi'(i,j) = 0.5\phi(i,j)$ in order to include the previously obtained sign value. Therefore, the Formula 11 may be represented as:

Page 16, lines 1-7:

When searching the fixed codebook in the above process, most of the computations are required in searching a position index of the optimal pulse in a loop of each track. Therefore, the high-speed search method of the present invention arranges values of each d'(n) in the tracks (t_0, t_1, t_2) and then searches ~~an~~ a position index which has the biggest d'(n) value among the three loops. ~~The~~ Tables 2 and 3 show ~~such~~ examples of the high-speed search method, including a search for specific sub-frames, which ~~follows~~ follow the below methods.

Page 16, lines 8-19:

At first, the position indexes of the tracks (t_0, t_1, t_2) are arranged in a descending order according to the d'(n) value. ~~As seen in the Table 4, a~~ Then, the position index that has the biggest probability to be an optimal pulse position, as shown in Fig. 4, is searched first. Because the numerator of the Formula 8 based on the d'(n) value is in a square type, its attribution is more than that of the denominator. A pulse position, which maximizes the correlation value ~~(Ck)~~ (C_k), has great possibilities to be an optimal pulse position. This can be easily understood ~~with the~~ from Table 4, which statistically shows probability to be selected as an optimal position for each pulse in the fixed codebook, arranged in a descending order according to the d'(n) value. In other words, a pulse position having the biggest d'(n) value is most probably an optimal pulse position.

Page 16, line 20 to Page 17, line 4:

Then, because the threshold value in the Formula ~~17~~ 15 is composed of only the d'(n) values, *i.e.*, the correlation vectors between the object signals and impulse response of the composite signals for each of the tracks (t_0, t_1, t_2), as described above, and arranged with the d'(n) values in a descending order, after calculating each d'(n) value of the tracks (t_0, t_1, t_2) and then determining whether ~~their~~ the sum of the d'(n) values is over the predetermined threshold value, the search process is executed if the sum is over the threshold value by the codebook search is finished if the sum is not over the threshold value.

Page 17, lines 5-8:

As described above, the candidate values over the threshold may be searched in a high-speed by sequentially arranging the fixed codebook according to the d'(n) values and calculating the correlation value ~~Ck~~ C_k on the basis of the arranged codebook.

Page 19, lines 1-4:

The step of arranging the pulse position indexes of the tracks (t_0, t_1, t_2) according to the correlation value of each track T110 ~~is~~ involves comparing sizes of correlation ~~vectors~~ values of each pulse position index for each track and then arranging them in a descending order.

Page 19, lines 5-9:

In other ~~word~~ words, the step T110 compares the correlation ~~vector sizes~~ value magnitudes obtained for all pulse position indexes of the track 1 (t_0) and then arranges the correlation ~~vectors~~ values in a descending order. The step T110 executes an arrangement for the tracks 1 and 2 in a descending order by using the same ~~way~~ approach.

Page 19, lines 10-12:

Table 3 is a chart showing the process of arranging the pulse position indexes in a descending order according to the correlation ~~vector sizes~~ value magnitudes of each of the tracks (t_0, t_1, t_2) in a specific sub-frame.

Page 19, lines 13-16:

Referring to Tables 2 and 3, Table 2 ~~is assumed~~ assumes that the correlation value is given for each pulse position index and Table 3 shows pulse positions (or position indexes) arranged in a descending order on the basis of the correlation value.

Page 21, lines 19 to Page 22, line 3:

~~FIG.~~ As explained above, Table 4 is a chart showing statistical probabilities that each pulse position for the tracks 0, 1 and 2 is selected as an optimal pulse position for the tracks 0, 1 and 2 is selected as an optimal pulse position. As shown in the ~~figure~~ table, probability values that each pulse position for the tracks 0, 1 and 2 is selected as an optimal pulse position are arranged sequentially. Their arrangement is identical to that which is arranged in a descending order based on the size of the correlation value for each pulse position index.

Page 22, lines 8-11:

Therefore, the pulse position, which maximizes the correlation value ~~(Ck)~~ (C_k), is very probable to be the optimal pulse position, while the pulse position having the biggest correlation vector size is most probable to be the optimal pulse position.

Page 24, lines 3-9:

The present invention gives effects of reducing computational ~~complex~~ complexity required to search the codebook without signal distortion in quantizing the LSP ~~counts~~ parameters of the speech encoder using SVQ manner, and reducing computational ~~complex~~ complexity without loss of tone quality in G.729 fixed codebook search by performing candidate selection and search on the basis for the correlation value size of the pulse position index.